Comput. Math. Applic. Vol. 15, No. 4, pp. 333–336, 1988
Printed in Great Britain.

0097-4943/88 $3.00 + 0.00

COMPUTERS, TREES AND ABELIAN GROUPS

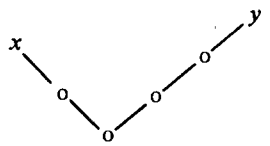
F. RICHMAN
New Mexico State University, Las Cruces, NM 88003, U.S.A.

Abstract—The problem of classifying pairs consisting of a finite Abelian group and a subgroup leads to the study of rooted trees whose nodes are decorated with natural numbers that strictly increase as you go towards the root. The lattices of trees that correspond to indecomposable pairs that are bounded by p^n were generated by computer up to $n = 6$. As a result, an unexpected, almost complete, duality was discovered in these lattices.

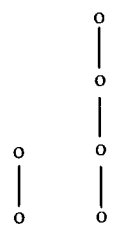
The structure of a finite Abelian group is completely described by simple numerical invariants: each such group can be written as a direct sum of cyclic groups of orders $n_1, n_2, \ldots, n_m$ such that $n_1 > 1$ and n_{i+1} is a multiple of n_i for $i = 1, \ldots, m-1$. The numbers n_i are unique even though the group can be written as a direct sum of cyclics in many different ways.

A much more complicated problem is to classify the subgroups of a finite Abelian group, paying attention not only to the structure of the subgroup, but how it sits inside the group. As is the case for finite Abelian groups themselves, this problem can easily be reduced to the study of *p-groups* —groups in which each element has order a power of p, for some fixed prime p. Henceforth, *the word "group" will refer to a finite p-group.*

Every rooted tree can be thought of as a presentation of a p-group, via generators and relations, as follows: the generators are the nodes of the tree, we set $pr = 0$ for the root r, and we set $px = y$ if the node y sits just below the node x in the natural picture of the tree with the root at the bottom. For example, the tree

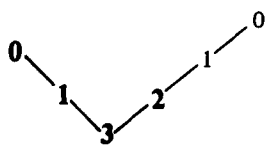


has two leaves x and y; the other nodes are identified in the group with px, p^2x, py and p^2y. The root is $p^2x = p^3y$, so the group can also be described as having generators x and y subject to the relations $p^2x = p^3y$ and $p^3x = 0$. This group is a direct sum of a cyclic group of order p^2 and a cyclic group of order p^4, and admits another presentation in terms of the forest



whose leaves may be taken to be the elements $x - py$ and y.

Although the classification problem for arbitrary subgroups is extremely difficult—there is evidence, which we won't go into here, that it is probably unsolvable—we can hope to classify special kinds of subgroups. The class of subgroups that motivated the work described in this paper, the details of which can be found in Ref. [1], are those generated by a subset of the nodes of some forest-presentation of the group. Any subgroup of a cyclic group has this property, as does any direct summand of an arbitary group. The simplest nontrivial example is



where we decorate the nodes with numbers indicating how far down in the tree they are; this is, perversely but traditionally, called the *height* of the node. The nodes generating the subgroup are in boldface. It turns out that essentially all the information is contained in the subtree, together with the heights of its nodes in the original tree. This gadget is called a *valuated tree*. For typographical reasons we write it as

$$\begin{array}{ccc} 0 & & 0 \\ 1\ 2 & \text{or} & 1\ 2 \\ 3 & & 3 \end{array}$$

the latter being how our computer prints them out. For the purposes of this paper a *valuated tree* is a finite rooted tree with nonnegative integers, called *values*, on each node, so that the values increase strictly as you proceed towards the root. Different nodes may have the same value.

If H is a subgroup of a group G, and G can be decomposed in a nontrivial way as a direct sum $G_1 \oplus G_2$ such that $H = H_1 \oplus H_2$ with $H_i \subseteq G_i$, then we say that the pair $H \subseteq G$ is *decomposable*. It suffices to classify indecomposable pairs, and it was shown in Ref. [2] that a valuated tree T gives an indecomposable pair if and only if T is irretractible. This latter notion is defined in terms of a *map of valuated trees*, which is a function f that takes roots to roots, weakly increases values, and such that $f(y)$ sits just below $f(x)$ if y sits just below x. A *retraction* of a valuated tree T is a map f from T to T such that $f^2 = f$; a tree is *irretractible* if the identify is its only retraction. The tree

$$\begin{array}{c} 0 \\ 1\ 2 \\ 3 \end{array}$$

is irretractible, while the tree

$$\begin{array}{c} 0 \\ 2\ 1 \\ 3 \end{array}$$

admits the retraction that takes the node of value 1 to the node of value 2, and fixes all other nodes.

The irretractible trees form a lattice under the partial ordering in which $T_1 \leqslant T_2$ means there is a map from T_1 to T_2. If we restrict ourselves to irretractible trees whose values do not exceed n, we get a finite lattice $\mathscr{T}_n$. The lattice $\mathscr{T}_3$, which consists of 16 trees, is shown in Fig. 1.

A program was written in C on a VAX 11/750 to generate these lattices $\mathscr{T}_n$. Each tree in the lattice is represented by the value of its root, pointers to the trees that constitute its branches, and pointers to the trees that are its immediate predecessors in the lattice. These latter pointers not only provide useful information when investigating the structure of the lattice, but are used in the generation process itself, which is done one level at a time.

The trees in each level are the immediate successors of the trees in the previous level. There are two kinds of immediate successors of a tree. The simple kind is constructed by raising the value of the root by 1, if possible; the others are generated as follows. Let B be the set of branches of the tree. Construct the set B^* of trees that are the minimal elements of $\{T$: there is no $S \in B$ such that $T \leqslant S\}$. This is done inductively by first choosing a level that is greater than the level of any tree in B, and flagging all the trees in that level. Given a flagged level, we put in B^* each flagged tree whose immediate predecessors all lie in B, and we flag any tree in the previous level which is not in B, and is an immediate predecessor of some flagged tree. Having constructed B^*, for each $T \in B^*$ whose root value is less than the value m of the root of the given tree, we construct a tree whose root has value m and whose branches consist of T together with those branches of the given tree that do not lie below T. The proof that this construction is correct may be found in Ref. [1].

The lattice $\mathscr{T}_4$ contains 43 trees and can be constructed by hand, although it is easy to make mistakes. The lattice $\mathscr{T}_5$ has 217 trees and can be computed fairly quickly by machine. The first really challenging problem for our program was $\mathscr{T}_6$. To monitor the progress of the computation, upon completion of the generation of a level we printed out the number of trees in it. The numbers slowly increased: level 15 had 14 trees, level 95 had 577 trees. We awaited the point where the numbers would start decreasing; the levels were taking a few minutes each to generate. We had

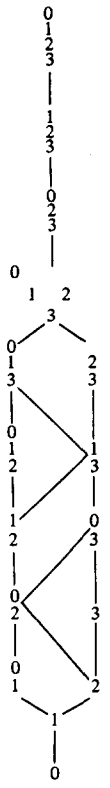


Fig. 1

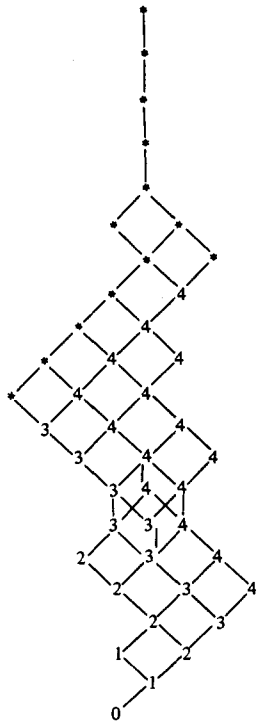


Fig. 2

no idea how many levels there would be, or whether we would even be able to compute all of $\mathscr{T}_6$. Actually we could have figured out how many levels were in $\mathscr{T}_6$ from how many trees were in $\mathscr{T}_5$, but our theoretical work had not caught up with our experimental work as yet. After 577 the number of trees in the levels started to decrease, and we happily anticipated a conclusion to the computation. But then the numbers started increasing again.

We noticed that level 119 also had 577 trees. Upon closer examination we saw a perfect symmetry centered at level 107. When the computation ended, at level 223, we found that the symmetry extended from level 6 to level 208. The symmetry of the numbers suggested a symmetry in structure, so we wrote a program to find a lattice duality in the levels between 6 and 208. This duality was found, was unique, and extended uniquely to a duality that included all levels from 0 to 208, but left out 17 trees in levels 209 through 223. Figure 2 is a picture of this partial duality in $\mathscr{T}_4$. The trees participating in the lattice are labeled with the values of their roots.

As you can see from Fig. 2, the partial duality might have been conjectured from hand computation of $\mathscr{T}_4$, but the fact is that it was not. One reason it was missed is that it is not a full duality, and the number of nonparticipating trees is a substantial percentage of the total number of trees. On the other hand, the numerical evidence from $\mathscr{T}_6$ was so striking that even clods like us could see that something was going on.

The duality remained mysterious for some time, and in some sense is still mysterious. An inductive construction of the duality is provided in Ref. [1], but not much insight is provided thereby. There seems to be no natural connection of a tree with its dual, nor could we find some grander duality in Abelian group theory of which this duality is a reflection.

The existence of the duality was suggested by the computer, and the general inductive construction of Ref. [1] was greatly facilitated by comparing the computer-constructed dualities in $\mathscr{T}_4$ and $\mathscr{T}_5$. The computer also improved the proof of the inductive construction. In order to check that the inductive construction was really correct, a program was written to compute the duality in $\mathscr{T}_5$ from the duality in $\mathscr{T}_4$ by the inductive construction, and compare it with the known duality in $\mathscr{T}_5$. There was one anomalous case in going from $\mathscr{T}_4$ to $\mathscr{T}_5$ which was temporarily left

out of the program on the theory the check would be better if it found that we were off by one rather than just saying that everything was O.K., and it was a bit of a pain to include the case. Upon running the program, however, the report was that everything was O.K. After trying unsuccessfully to debug the program, the program was discovered to embody a more elegant formulation of the construction than the proof, and there was no anomalous case from its point of view. This resulted in a substantial simplification of the proof.

The fact that $\mathscr{T}_6$ has 577 trees at level 119 means that $\mathscr{T}_7$ contains over 2^{577} trees, because every subset of the level 119 trees, when attached to a root of value 7, gives a distinct tree in $\mathscr{T}_7$. Thus, $\mathscr{T}_6$ is the last lattice that can be computed in its entirety. We did do some partial computations on $\mathscr{T}_n$ for $n = 7, \ldots, 12$ in order to verify the duality at the top and bottom of these lattices; this required another algorithm to compute the lattices from the top down. Thinking about this top–down algorithm aided in the discovery of the inductive proof of the duality.

REFERENCES

1. D. Beers, R. Hunter, F. Richman and E. Walker, Computing valuated trees. *Proc. 1985 Oberwolfach Conf. on Abelian Groups*. Gordon & Breach, New York (1987).
2. R. Hunter, F. Richman and E. Walker, Simply presented valuated Abelian p-groups. *J. Algebra* **49**, 125–133 (1977).